VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practise in the Company

2010 Antonín Míček

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Antonín Míček**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe
Individual Professional Practise in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: PEGATRON Czech s.r.o.
2. Struktura závěrečné zprávy:
   a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
   b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
   c) Zvolený postup řešení zadaných úkolů.
   d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
   e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
   f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Eliška Ochodková**

Konzultant bakalářské práce: Wladyslaw Motyka

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA * FAKULTA elektrotechniky a informatiky * OSTRAVA *

doc. Dr.Ing. Eduard Sojka
vedoucí katedry

prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě 14. dubna 2010 . . . . . .

PODĚKOVÁNÍ

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Především kolegům v práci, kteří vždy byli ochotni poradit a pomoct.

ABSTRAKT

Tato bakalářská práce má za cíl popsat průběh absolvování individuální odborné praxe na oddělení PE (Product Engineering) firmy PEGATRON. Celá práce je zaměřená na rozšíření programu WGET o další funkce a na zlehčení manipulace s Hi-Potem na výrobních linkách, jelikož právě touto činností jsem se v průběhu celé své odborné praxe zabýval.

V úvodní kapitole práce je popsáno zaměření firmy a mé pracovní zařazení. V dalších kapitolách pak pokračuji v popisování svých zadaných úkolů a postupu při jejich řešení. Nakonec je celá praxe zhodnocena z pohledu nabitých znalostí a dovedností a celkovým přínosem pro mě samotného.

KLÍČOVÁ SLOVA

WATTCP, WGET, ping, tcpinfo, cURL, MD5, Hi-Pot, python, wxPython

ABSTRACT

This bachelor thesis is intended to describe the sequence of the completion of individual professional experience on department PE (Product Engineering), PEGATRON. All work is aimed at expansion WGET of additional functions and debunkery handling Hi-Pot on production lines, as it is the business I have throughout my professional experience dealt with.

In the introductory chapters of my thesis is described the orientation of the company and my work position. In the following chapters I continue describing my tasks, chosen procedures and solutions. Finally, the whole experience is evaluated in terms of acquired knowledge and skills and the overall benefit for myself.

KEYWORDS

WATTCP, WGET, ping, tcpinfo, cURL, MD5, Hi-Pot, python, wxPython

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PE	-	Product engineering
DOS	-	Disk Operating System
DJGPP	-	DJ's Gnu Programming Platform
GCC	-	GNU Compiler Collection
WGET	-	World Wide Web and get
TCP/IP	-	Transmission Control Protocol / Internet Protocol
HTTP	-	Hypertext Transfer Protocol
FTP	-	File Transfer Protocol
PING	-	Packet InterNet Groper
DNS	-	Domain Name System
MAC	-	Media Access Control
DHCP	-	Dynamic Host Configuration Protocol
UDP	-	User Datagram Protocol
CURL	-	Client for URLs
MD5	-	Message-Digest algorithm 5
HI-POT	-	High-Potential
RS 232	-	Recommended Standard 232
STB	-	Set-top-box

OBSAH

1. ÚVOD

Má odborná praxe proběhla ve společnosti PEGATGON Czech s.r.o. Během ní jsem byl postaven před několik úkolů. Týkaly se rozšíření programu WGET o další funkce a na zlehčení manipulace s Hi-Potem na výrobních linkách.

V práci je popsaná historie a současnost firmy PEGATRON a oddělení PE, krátký náhled na zodpovědnost a pravomoci našeho oddělení. V kapitole 2 popisuji jednotlivé úlohy a problémy s kterými jsem byl po čase praxe konfrontovaný. V první podkapitole začínám rozšířením programu WGET, kde charakterizuji jednotlivé popisy a implementace funkcí, které jsem měl do programu WGET přidat. V další podkapitole 2.2 popisuji seznámení s Hi-Potem, komunikaci Hi-Potu s PC přes RS232, kompletní řešení daného úkolu a popis odzkoušení v praxi. V závěrečných kapitolách jsou pak popsané získané a chybějící znalosti a schopnosti k vyřešení jednotlivých úloh a problémů a v závěru celkové zhodnocení praxe.

1.1. Zaměření firmy PEGATRON Czech s.r.o.

Společnost PEGATRON Czech s.r.o., která sídlí v průmyslové zóně Ostrava – Hrabová, je součástí nadnárodní korporace PEGATRON/UNIHAN Corp. se sídlem v Taipei na Taiwanu. Tato společnost patří mezi největší světové společnosti v oblasti informačních technologií, mezi jehož nejznámější produkty patří základní desky počítačů, servery a síťové prostředky, notebooky, kapesní počítače, multimedia a další.

PEGATRON Czech s.r.o. byla založena v roce 2002 jako ASUS Czech, kdy působila v Rudné u Prahy a zabývala se zejména výrobou stolních počítačů. V roce 2004 byla společnost oceněna vládou ČR jako investor roku. Dále v roce 2004 začala výstavba vlastního výrobního závodu v nově založené průmyslové zóně v Ostravě-Hrabové a od ledna 2005 se veškerá výroba koncentrovala v Ostravě. V červnu 2005 se v závodě začaly poskytovat servisní činnosti pro zákazníky z celé Evropy.

Předmětem činnosti PEGATRON Czech je především kompletace osobních počítačů a dalších výrobků elektrotechnického průmyslu, a to na základě individuálních požadavků zákazníka. V zákaznickém centru v Ostravě jsou také vykonávané servisní služby klientů z Evropy, Asie a Severní Afriky, kteří používají naše produkty a také i produkty ASUS. PEGATRON Czech je nositelem certifikátu ISO 9001, ISO 14001 a OHSAS 18001.

Společnost má v současnosti 1150 kmenových zaměstnanců a prostřednictvím personálních agentur zaměstnává také několik stovek dočasně přidělených zaměstnanců. Spolupracuje se středními školami technického zaměření a s VŠB-TU, především FEI, jejímž absolventům nabízí možnost uplatnění nabytých vědomostí a získání zkušeností z práce ve velké nadnárodní společnosti. Aktivně

se podílí na ekonomickém i sociálním rozvoji regionu Severní Morava. Společnost je členem IT Clustru, jejichž posláním je zajistit přípravu lidských zdrojů, vytvořit potenciál pro řešení inovačních projektů a zajistit společné aktivity v oblasti marketingu [1].

1.2. Pracovní zařazení studenta

Jako praktikant jsem byl zařazen na oddělení PE (Product engineering) ve firmě PEGATRON jako technik SW podpory výroby. Tohle oddělení podporuje výrobní proces, zavádí nové projekty, řeší funkční problémy na lince, kontroluje správnost nastavení testovacích programů pro různé projekty.

Mým úkolem v rámci firmy bylo podílet se na SW podporu výrobního procesu, vytváření programů sloužící k jeho podpoře, sbírání dat z testovacího procesu a následné doplnění chyb z provozu.

1.3. Zadané úkoly

V průběhu 2 semestrů jsem měl za úkol následující body:

1. Rozšíření programu WGET
    - přidat funkci PING
    - přidat funkci TCPINFO
    - přidat funkci upload
    - přidat funkci MD5 pro download a upload
2. Zlehčení manipulace Hi-Potu na výrobních linkách
    - seznámit se s Hi-Potem
    - naprogramovat testovací komunikaci v jazyku python
    - kompletní komunikaci Hi-Potu s PC přes RS232 v grafickém prostředí
    - odzkoušení programu v praxi

Cílem prvního úkolu bylo zpracování programu pro odesílání a přijímání dat po síti v prostředí MS DOS. Tento program měl umožňovat využívání zmíněných funkcí v rámci jednoho programu a měl být využíván zejména ke stahování programů při instalaci či testování nových počítačových sestav.

Druhý úkol měl především zvýšit a zabezpečit kvalitu procesu výroby, zpřehlednit práci na pracovišti se zařízením Hi-Pot a zajistit větší kontrolu na dodržování pracovních povinností operátorů.

2. ŘEŠENÍ ÚKOLŮ

2.1. Rozšíření programu WGET

Základní požadavky na spuštění programu WGETu:

1. FreeDOS
2. DJGPP (moje verze 2.03)
3. WATTCP (moje verze 2.2 dev.rel.10)

Nejprve jsem na počítač nainstaloval FreeDOS, volný DOS-kompatibilní operační systém.

Potom následovalo instalování DJGPP. Jedná se o 32-bitový C/C++/ObjC/Ada/Fortran vývojářský balík právě běžící pod DOSem. Používal jsem ho spíše jako kompilátor GCC, jako překladač programovacího jazyka C [3].

A na závěr přišla nejdůležitější věc – WATTCP. Knihovna WATTCP je jednou z knihoven pro programování TCP/IP pro operační systém DOS, které jsou k dispozici jako public domain včetně zdrojových textů [4].

2.1.1. Samotný WGET

Po nainstalování všech tří požadavků jsem začal spouštění samotného WGETu (moje verze 1.11.4). WGET je počítačový program, který slouží jako jednoduchý a výkonný stahovač souborů. Implementuje přenos souborů přes protokoly HTTP a FTP. Zdrojové kódy programu jsou ke stažení bezplatně na internetu [5].

Po stažení programu samozřejmě přišla kompilace. Při kompilaci daného programu se vyskytlo pár chyb. Musel jsem:

1. vytvořit chybějící zdrojový kód – version.c,
2. přejmenovat správně pár hlavičkových souborů,
3. přidat do zdrojového kódu *wchar.h* chybějící funkce,
4. odstranit dvě funkce v kódu *sysdep.h*, které již byli jednou deklarovány,
5. odstranění v kódu *main.c* pár proměnných a funkce, které tyto proměnné používají.

Po odstranění všech daných chyb kompilace proběhla úspěšně. Nyní samotný program sloužil ke stahování souborů ze serveru – *wget <adresa>*.

2.1.2. Implementace funkce PING

Po spouštění základního programu WGET přichází implementace první funkce – PING. Program PING umožňuje prověřit funkčnost spojení mezi dvěma síťovými rozhraními (počítače, síťová zařízení) v počítačové síti, která používá rodinu protokolů TCP/IP. Ping při své činnosti periodicky odesílá IP datagramy a očekává odezvu protistrany. Při úspěšném obdržení odpovědi vypíše délku zpoždění a na závěr statistický souhrn [2].

Parametrem programu PING je doménové jméno nebo IP adresa síťového rozhraní, jehož dostupnost chceme prověřit. Je-li uvedeno doménové jméno, je nejprve přeloženo pomocí DNS na IP adresu. Výzvy jsou odesílány na cílovou IP adresu a ve stanoveném limitu se očekává odpověď. Jednotlivé výzvy obsahují čísla, podle kterých je možné identifikovat jednotlivé odpovědi nebo jejich ztrátu. Program průběžně vypisuje, které odpovědi již došly a s jakým zpožděním [2].

Zdrojový kód PINGu jsem vzal z knihovny WATTCP. K danému zdrojovému kódu stačilo vytvořit hlavičkový soubor – ping.h, který obsahoval deklarace proměnných a funkcí. Potom do programu WGET se přidala nová funkce ping a to připojením do základního zdrojového kódu vytvořený hlavičkový soubor – *#include <ping.h>*. Nyní po kompilaci WGETu můžeme používat nový parametr, který spustí funkci ping – *wget –ping <ip_adresa>*. Toto řešení funguje jenom pro parametr PINGu, když zadáváme IP adresu, ke které chceme zjistit dostupnost. Když zadáme parametr doménové jméno, dostupnost neprověříme.

Pro konečné řešení tedy vytvoříme novou funkci do zdrojového kódu – host.c – v programu WGET, která bude z doménového jména vracet její IP adresu ze seznamů hostů. Teď stačí jenom volat naší příslušnou funkci v základním zdrojovém kódu WGET, když uživatel zadá doménové jméno. Po nové kompilaci můžeme používat jako parametr PINGu i doménové jméno – *wget –ping <ip_adresa_nebo_domenove_jmeno>*.

2.1.3. Implementace funkce TCPINFO

Po úspěšné implementace první funkce přichází implementace další funkce – TCPINFO. Program TCPINFO zobrazuje základní informace TCP – MAC adresu, IP adresu, síťovou masku, IP adresy bran, jméno serveru, velikost TCP a UDP socketu, atd.

Postup je podobný jako u implementace PINGu. Zdrojový kód TCPINFO jsem také vzal z knihovny WATTCP. K danému zdrojovému kódu zase stačilo vytvořit hlavičkový soubor – tcpinfo.h, který také obsahoval deklarace proměnných a funkcí. Potom do programu WGET se přidala nová funkce tcpinfo a to připojením do základního zdrojového kódu vytvořený hlavičkový soubor –

#include <tcpinfo.h>. Nyní po kompilaci WGETu můžeme používat nový parametr, který spustí funkci tcpinfo – *wget –tcpinfo*. Na obrazovku se nyní objevili příslušné informace TCP. Ve firmě chtěli jenom zobrazení tří funkcí TCP – MAC adresu, IP adresu a IP adresy brán. Následně spouštěním této funkce se vytvořil soubor *dhcp-ip*, který obsahuje informace o přidělení IP adresy pomocí DHCP, vypršení IP adresy, atd.

2.1.4. Implementace funkce UPLOAD

Po základních funkcí PING a TCPINFO dochází k rozšíření WGETu o nahrávání (upload) souborů na server. K tomu potřebujeme program cURL (moje verze 7.19.7.). cURL je nástroj pro přenos dat po protokolech jako HTTP, FTP a dalších. Obsahuje konzolový program a knihovnu – libcurl – implementující všechny funkce programu. Zdrojové kódy programu jsou ke stažení bezplatně na internetu [6].

Pro samotnou implementaci nám stačí knihovna libcurl a zdrojový kód – httpput.c – který obsahuje příslušný upload. Nyní jsme postupovali úplně stejně jako v předchozích případech. K danému zdrojovému kódu – httpput.c – zase stačilo vytvořit hlavičkový soubor – httpput.h, který také obsahoval deklarace proměnných a funkcí. Potom do programu WGET se přidala nová funkce upload a to připojením do základního zdrojového kódu vytvořený hlavičkový soubor – *#include <httpput.h>*. Nesmíme ale zapomenout na připojení knihovny – libcurl – do kompilačního souboru *makefile.dj*. Nyní po kompilaci WGETu můžeme používat nový parametr, který spustí funkci upload – *wget –upload <adresa>*.

2.1.5. Implementace MD5 pro download a upload

Poslední funkcí je implementace MD5 pro download a upload v programu WGET. MD5 je rozšířená hašovací funkce, která vytváří otisk o velikosti 128 bitů. MD5 je zaměstnána v celé řadě bezpečnostních aplikací, a je také běžně používaná pro kontrolu integrity souborů. Hašovací funkce MD5 je typicky vyjádřena jako 32místné hexadecimální číslo [2].

K implementaci stačilo stáhnout 4 zdrojové kódy – md5.c, md5.h, mddriver.c, mddriver.h. Začneme implementací MD5 pro download a to tak, že do zdrojového kódu – http.c - programu WGET se přidá nová funkce md5 a připojí se hlavičkový soubor – *#include <mddriver.h>*. Nyní stačí zavolat příslušnou funkci md5 ve funkci *http_loop*.

Implementace pro upload je jednodušší než pro download. Tady stačilo připojit do základního zdrojového kódu od programu WGET hlavičkový soubor – *#include <mddriver.h>*. A ve vytvořené funkci upload volání funkce md5 a přiřazení do volané adresy výsledek funkce md5.

Nyní při stahování a nahrávání souborů na server jsme měli nastaveno zabezpečení aplikace a kontrolu integrity daných souborů.

2.1.6. Konečné opravy WATTCP

Po všech předešlých implementací je daný úkol splněn a rozšířený WGET funkční. V praxi pak nastaly dvě chyby:

1. Při selhání přidělování IP adresy přes DHCP se nesměla IP adresa přidělit přes RARP
2. Náhodné zamrznutí po přidělení IP adresy přes DHCP

První chyba se odstranila přepsáním hodnoty – *try_rarp na FALSE* – ve zdrojovém kódu od WATTCP – sock_ini.c – ve funkci *tcp_do_boot*.

Druhá chyba se odstranila přidáním jedné podmínky, ve které automaticky po přidělení IP adresy skončí DHCP, ve zdrojovém kódu od WATTCP – pcdhcp.c – ve funkci *dhcp_fsm*.

Po odstranění předešlých chyb program WGET odzkoušen v praxi a nyní je používán ve firmě PEGATRON Czech s.r.o.

2.2. AUTO Hi-Pot

Obrázek 1: CHROMA ATE 19032

Druhý projekt se zabýval zlehčením manipulace s Hi-Potem na výrobních linkách při testování STB. Používaný Hi-Pot na linkách je od technologie CHROMA ATE 19032 [obr.1].

2.2.1. Seznámení s Hi-Potem a volba programovacího jazyka

CHROMA 19032 je určena pro rychlé a snadné výrobní testování jednofázových elektronických přístrojů a nástrojů. Je schopna provést 5 základních elektronických zkoušek bezpečnosti.

První funkcí je AC mód. Provádí test přes napětí v rozmezí od 50 do 5000V a zbytkový proud v rozmezí od 1µA do 40mA. Tenhle mód je ideální pro testování výrobků a nástrojů s velkým zbytkovým proudem. Další funkce DC mód testuje přes napětí v rozmezí 50 do 6000V se zbytkovým proudem do 12mA. Používá se pro rychlé nabíjení kapacitního zařízení nebo produktů. Třetí funkce, IR mód, vypočítává a zobrazuje izolační odpor výrobku v ohmech. Tento odpor měří v rozsahu od 1MΩ do 50GΩ s napětím od 50 do 1000V. Následující funkce, GB mód, se používá pro ověření integrity uzemněných systémů s proudem v rozsahu od 1 do 30A a odporu mezi 10mΩ a 510mΩ. Posledním módem je OSC, který řeší problémy s nepřipojenými kabelami k výrobkům.

Ve společnosti PEGATRON se implementují pouze 3 funkce: DC, IR a OSC.

CHROMA 19032 hlavně využívá programovací jazyk BASIC nebo python. Vybral jsem si programovací jazyk python, protože na oddělení PE více lidí s ním umí pracovat.

2.2.2. Testovací komunikace Hi-Potu

Prvním programem byl základní testovací, který vyzkoušel komunikaci mezi Hi-Potem a PC přes RS232 (COM port). Program zjistil aktuální nastavení Hi-Potu, smazal aktuální nastavení, vložil do Hi-Potu nové nastavení (módy AC a IR), otestoval dané funkce a vytisknul na obrazovku nové aktuální nastavení Hi-Potu.

Testovací program se nachází v manuálu daného Hi-Potu v programovacím jazyku BASIC. První odzkoušení bylo přes jazyk BASIC, jenom zkopírování kódu, které proběhlo úspěšně. Druhé odzkoušení bylo už přepsané do jazyka python, které taky proběhlo úspěšně.

2.2.3. Popis kompletního řešení úkolů

Při spouštění programu se nejdříve načte (naskenuje) sériové číslo Hi-Potu. Podle daného sériového čísla se z konfiguračního souboru *hipot_sn.ini* zjistí, zda daný Hi-Pot bude používat funkci Hi-Potu OSC. Podle toho se zobrazí všechny modely STB z konfiguračního souboru *config.ini* bez funkce OSC nebo s OSC [obr.2]. Po vybrání modelu se z *config.ini* nastaví Hi-Pot podle daného modelu, např. u vybrání modelu TDS865NS se nastaví funkce IR, a u vybrání modelu TDS850NS funkce AC a IR.

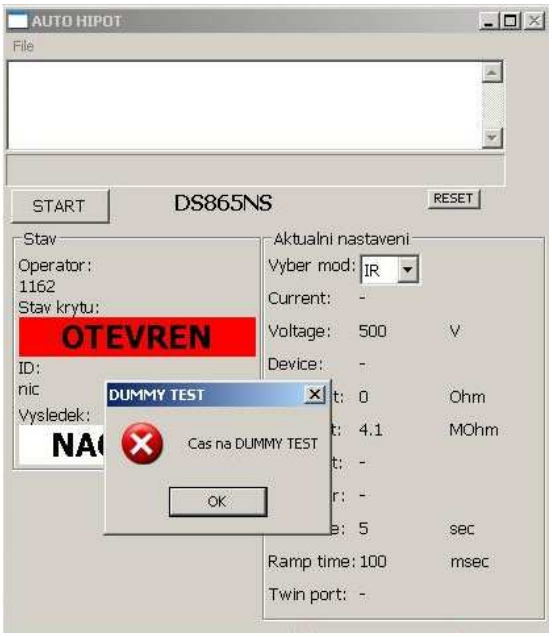
Obrázek 2: Modely STB

Následuje DUMMY test, zda daný Hi-Pot testuje správně [obr.4]. Odkazuje se na hlavní konfigurační soubor *settings.ini*, ve kterém jsou informace o DUMMY testu [obr.3]. Program nejdříve zkontroluje proměnnou *dummy* v *setting.ini*, zda je nastavena na ON nebo OFF. Jedná se o proměnnou, která určuje, zda se bude provádět kontrola Hi-Potu nebo ne. Následuje další kontrola proměnné *dummy_pass*, která může být taky nastavena na ON nebo OFF. Charakterizuje, zda se má otestovat DUMMY PASS. Naskenuje se sériové číslo DUMMY PASS. Program nyní kontroluje sériové číslo s další proměnnou *dummy_pass_isn*, která obsahuje správné sériové číslo DUMMY PASS. Když se naskenované číslo shoduje s touhle proměnnou, program pokračuje testováním. Neshoduje-li se, musí se načíst nové sériové číslo. Pokračujeme testováním DUMMY PASS. Jestli neprojde testem, otestuje se znovu. Projde-li, test proběhl úspěšně [obr.5]. Následuje kontrola proměnné *dummy_fail*, která může být taky nastavena na ON nebo OFF. Ta zase charakterizuje, zda se má otestovat DUMMY FAIL. Naskenuje se sériové číslo DUMMY FAIL. Program nyní kontroluje sériové číslo s další proměnnou *dummy_fail_isn*, která obsahuje správné sériové číslo DUMMY FAIL. Když se naskenované číslo shoduje s touhle proměnnou, program zase pokračuje testováním. Neshoduje-li se, musí se načíst nové sériové číslo. Pokračujeme testováním DUMMY FAIL. Jestli projde testem, otestuje se znovu. Neprojde-li, test proběhl úspěšně.

```
[DUMMY]
dummy=on

dummy_pass=on
dummy_pass_isn=111100UPC/V0008

dummy_fail=on
dummy_fail_isn=011100UPC/V0008

dummy_time=240
;dummy time v minutach
```

Obrázek 3: Nastavení DUMMY testu

Tenhle test funkčnosti se provádí vždy na začátku nebo podle nastavení poslední proměnné *dummy_time*. Po uplynutí daného času se zobrazí upozornění [obr.4].



Obrázek 4: Upozornění na DUMMY test

Obrázek 5: DUMMY PASS test

Po kontrole Hi-Potu se řeší, zda je zapnutá nebo vypnutá databáze.

Když je vypnutá databáze a zároveň Hi-Pot nepoužívá funkci OSC, pokračuje se načítáním sériového čísla STB, který chceme otestovat [obr.5]. Mezitím se kontroluje, zda kryt [obr.6], kde se nachází zapojený STB, je otevřený nebo ne. Když se náhle otevře, musí se načíst znova sériové číslo STB. Když je zavřený, Hi-Pot provádí test STB. Projde-li test úspěšně, STB prošel a pokračuje se načítáním dalšího sériového čísla STB. Neprojde-li test úspěšně, STB se může otestovat znovu [obr.5].

Když je vypnutá databáze a Hi-Pot používá funkci OSC, pokračuje se načítáním sériového čísla STB, podle kterého se nakalibruje Hi-Pot. Změří se napětí a kapacita STB a uloží se do paměti. Podle těch to hodnot se kontroluje u každého STB, zda STB má zapojené napěťové a zemnící kabely. Nyní zase pokračuje testování úplně stejně jako v předchozím případě, ale rozdíl je v tom, že se nyní bude i kontrolovat zapojení kabelů do STB v krytu.

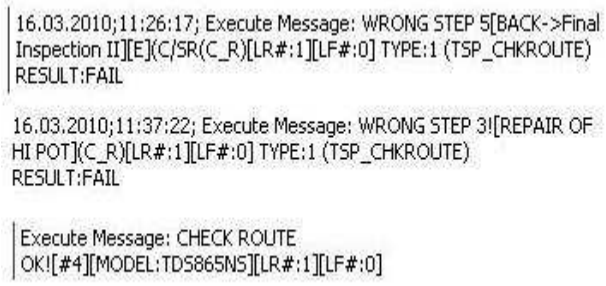
Obrázek 6: Testování STB

Obrázek 7: Kryt

Zapnutá databáze navíc řeší 3 spouštěcí utility.

První utilita je *tsp_login.exe*, která řeší přihlášení a odhlášení operátora. Operátor se přihlásí ještě před otestováním Hi-Potu (před DUMMY testem). Naskenuje svoje identifikační číslo a utilita zkontroluje, zda daný operátor existuje v databázi nebo ne. Jestli existuje, může testovat STB. Když ne, tak se musí přihlásit jiný operátor. Po vypnutí celé aplikaci, nebo tlačítka „Odhlášení“, utilita odhlásí daného operátora [obr.7].

Obrázek 8: LOGIN a LOGOUT



Obrázek 9: CHECKROUTE

Druhá utilita, *tsp_chkroute.exe*, řeší, zda testovaný STB je na správné pozici a zda STB odpovídá danému modelu. Kontroluje to hned po načtení sériového čísla STB. Neprojde-li utilitou, nebude daný STB otestován [obr.8].

Poslední utilita je *tsp_testresult*, která posílá do databáze výsledek testu. Utilita se spouští po otestování Hi-Potem. Když testování proběhlo úspěšně, utilita uloží daný STB do databáze bez chyby. Neproběhne správně test, operátor si určí, zda utilita uloží daný STB do databáze s chybou nebo bude STB testovat znovu pomocí RETESTU [obr.5].

Program obsahuje i administrátorskou část, kde může administrátor vidět všechny modely a jejich nastavení z konfiguračního souboru *config.ini*. Dále vytváří nový model s nastavením příslušných funkcí do *config.ini* a odesílá nastavení Hi-Potu vybraného modelu. Má také možnost i ruční kalibraci.

Celý program je napsaný v programovacím jazyku python a jeho grafickým prostředí wxPython. Řídí se podle 3 konfiguračních souborů. Nejdůležitější konfigurační soubor je *settings.ini*, který obsahuje nastavení programu (cestu k ukládání logů, kontrolu modelu při utilitě *tsp_chkroute.exe*, délku sériového čísla Hi-Potu a STB), nastavení databáze (zda je zapnutá nebo vypnutá, heslo databáze, název chyby, …), nastavení testování Hi-Potu (zda se provede nebo neprovede, zda se bude

testovat jenom DUMMY PASS, DUMMY FAIL nebo oba, a jejich sériové čísla) a nastavení sériových portů Hi-Potu a krytu. Další konfigurační soubor *config.ini* obsahuje modely a jejich nastavené funkce Hi-Potu. Poslední *hipot_sn.ini* charakterizuje model, verzi a sériové číslo Hi-Potu a zda podporuje funkci OSC.

Samozřejmě, že program vytváří i logy. Vytváří *error.log*, kde se zapisují chyby utilit, otevření krytu, nesouhlas délek sériových čísel, chyby Hi-Potu a chyby při testování. Vytváří i *dummy.log*, ve kterém se ukládá výsledek DUMMY TEST a kalibračního kusu, a *succes.log*, kde se ukládají výsledky úspěšných i neúspěšných testů.

2.2.4. Odzkoušení v praxi

Jakmile program byl úspěšně naprogramován a všechny požadavky splněny, odzkoušel jsem ho v praxi na výrobních linkách. Otestoval jsem kolem 100 STB a všechno proběhlo úspěšně. Nyní je program používán na výrobní lince ve firmě PEGATRON Czech s.r.o.

3. ZÁVĚR

3.1. Využité znalosti

Během prvního semestru průběhu praxe jsem byl v kontaktu s programovacím jazykem C. Nutné tedy byly především znalosti tohoto programovacího jazyka. Základní seznámení s tímto jazykem jsem získal díky předmětu Programování v C,C++ ve třetím semestru studia na VŠB-TUO.

Obecně jsem samozřejmě využil i veškeré zkušenosti s programováním, které jsem nabyl během studia na VŠB-TUO. Programování se týkalo 18 předmětů během všech 6 semestrů, nepřímých znalostí tedy byla spousta.

3.2. Chybějící znalosti

Během druhého semestru průběhu praxe jsem byl v kontaktu s programovacím jazykem python, jeho grafickým rozhraním wxPython a zařízením Hi-Pot. Konkrétní znalosti tohoto programovacího jazyka a zařízení jsem ovšem neměl žádné.

Nejdříve jsem se musel seznámit se zařízením Hi-Pot, které sloužilo na výrobních linkách na testování STB – jak se s Hi-Potem ovládá a hlavně testuje, jaké obsahuje všechny módy (funkce) pro testování STB, a hlavně jak můžu komunikovat s Hi-Potem přes PC. K těmto informacím mi pomohl hlavně manuál k příslušnému Hi-Potu.

A programovací jazyk python a jeho grafické rozhraní wxPython, které sloužilo pro zlehčení manipulace s Hi-Potem na výrobních linkách, je implementován v jazyce C. Znalosti programovacího jazyka C jsem popsal výše, ale k pythonu bylo nutné prozkoumat různé funkce navíc, hlavně připojení na COM port kvůli komunikaci s Hi-Potem.

3.3. Dosažené výsledky a zhodnocení

Celý můj pobyt ve společnosti PEGATRON hodnotím jako velmi přínosný, viděl a zkusil jsem práce ve velkovýrobě a aktivity s tím spojené. Ač už práce s lidmi, spolupráce s více odděleními na projektech a problémech. Toto všechno mi dalo zkušenosti do další práce ve velké firmě. Tato práce mě naučila myslet jinak jak ve škole, kde se na každý problém můžu zeptat cvičícího nebo přednášejícího. Po čase praxe jsem musel zužitkovat všechny vědomosti nejen ze školy na vyřešení problémů. Jak jsem už nevěděl dále, vždy pomohli ochotní kolegové. Nakonec jsem mohl všechny moje úkoly prověřit v praxi. Touto zkušeností jsem zakončil moji individuální praxi ve společnosti PEGATRON.

4. SEZNAM POUŽITÉ LITERATURY

[1] PEGATRON. ***Profil společnosti*** [cit. 2009-10-12].

Dostupný z WWW: <http://www.pegatron.jobs.cz/>

[2] WIKIPEDIE. ***Encyklopedie*** [cit. 2010-04-15].

Dostupný z WWW: <http://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana>

[3] DJGPP. ***Instalační balík k DJGPP*** [cit. 2009-10-19].

Dostupný z WWW: <http://www.delorie.com/djgpp/>

[4] WATTCP. ***Zdrojové kódy k WATTCP*** [cit. 2009-10-19].

Dostupný z WWW: <http://www.bgnett.no/~giva/>

[5] WGET. ***Zdrojové kódy k WGET*** [cit. 2009-10-20].

Dostupný z WWW: <http://ftp.gnu.org/gnu/wget/>

[6] CURL. ***Zdrojové kódy k CURL*** [cit. 2009-11-09].

Dostupný z WWW: <http://curl.haxx.se/download.html>

[7] Electrical Safety Analyzer 19032, Version 1.2. ***Manuál k Hi-Potu***, January 2005, P/N A11 000601

[8] Python. ***Dokumentace k Python*** [cit. 2010-04-16].

Dostupný z WWW: <http://docs.python.org/>

[9] wxPython. ***Dokumentace k wxPython*** [cit. 2010-04-16].

Dostupný z WWW: <http://www.wxpython.org/onlinedocs.php>

5. SEZNAM OBRÁZKŮ

6. PŘÍLOHY NA CD

bakalářská_práce.doc – kompletní bakalářská práce v DOC formátu

bakalářská_práce.pdf – kompletní bakalářská práce v PDF formátu

wget.rar – kompletní program prvního úkolu

autohipot.rar – kompletní program druhého úkolu

vývojový_diagram_autohipot.rar – vývojový diagram druhého úkolu AutoHipot